system **160** includes a BDMV directory **162** that contains the PLAYLIST **164**, CLIPINF **166**, STREAM **168**, AUXDATA **170**, and BACKUP **172** directories. The PLAYLIST directory **164** contains the database files **174** for movie playlists. Each playlist has its own xxxxx.mpls file. The CLIPINF directory **166** contains the database files **176** for clips. The STREAM directory **168** contains audio and video stream files **178** (e.g. BDA V MPEG-2 transport streams) and the SSIF directory **180**, which contains stereoscopic interleaved files **182**. The AUXDATA directory **170** contains sound data files **184** and font files **186**. The BACKUP directory **172** contains copies of the "index.bdmv file", the "MovieObject.bdmv" file, all the files in the PLA YLIST directory **164** and all files in the CLIPINF directory **166**. The BDMV directory **162** also contains the index.bdmv file **188** and the MovideObject.bdmv file **190**. The index.bdmv stores information describing the contents of the BDMV directory. The MovieObject.bdmv file stores information for one or more Movie Objects.

[0104] The white paper also provides a simplified overview of the BD-ROM data structure, which is reproduced as FIG. **10**. The BD-ROM data structure **200** includes an Index Table **202** at the top-level of the data structure, which defines the Titles and the Top Menu of a BD-ROM disc. An Index Table is conceptually illustrated in FIG. **11**. The Index Table **202** contains entry points for all of the Titles and the Top Menu and has an entry to a Movie Object **222** and/or a BD-J Object **224** designated for "First Playback" **226**. When the disc is loaded, the player refers to the "First Playback" entry to determine the corresponding Movie Object or BD-J Object that shall be executed. The Index Table **202** also contains an entry **228** for "Top Menu" and an entry **230** for each title.

[0105] Referring back to FIG. **10**, the layer below the Index Table includes the Movie Objects **204** and the BD-J objects **206**. Each Movie Object is an executable navigation command program. The navigation commands can launch PlayList playback or another Movie Object enabling the authoring of a set of Movie Objects for managing playback of Playlists in accordance with a user's interaction and preferences. BD-J objects **206** perform similar functions, but are implemented as Java Xlet.

[0106] The Movie Playlist layer **208** includes a plurality of Movie Playlists. A Movie Playlist is a collection of playing intervals in the Clips. One such interval is referred to as a PlayItem and includes an IN-point and an OUT-point, each of which refers to positions on a time axis of the Clip. The manner in which MoviePlaylists index into the clips is illustrated in FIG. **12**. Each Movie Playlist **230** includes IN-points **232** and OUT-points **234** that index to positions in a Clip **236**.

[0107] Referring back to FIG. **10**, the bottom layer of the BD-ROM data structure is the Clip layer **210**. An AV stream file together with its associated database attributes is considered to be one object. The AV stream file is called a Clip AV stream file **212**, and the associated database attribute file is called a Clip Information File **214**. The Clip AV stream file **212** stores an MPEG-2 Transport Stream in accordance with ISO/IEC 13818-1 in a structure compliant with the BD-ROM A V specification. The Clip Information file stores the time stamps of the access point into the corresponding A V stream file. A player can read the Clip Information file to find out the position where it should begin to read the data from the A V stream file.

[0108] The Clip AV stream file can include an Interactive Graphics stream that is utilized in HDMV to generate a "Pop-Up" Menu Interface. In this case, video playback can continue while the HDMV Interactive Graphics are on the screen or video playback may be paused. Using the HDMV Interactive Graphics framework, multi-page menus can be defined with special commands available for inter-page navigation. As part of the framework, Button Objects **216** can be defined. When a Button is activated, a corresponding navigation command is executed which causes the display to change to a specified page. The HDMV Interactive Graphics framework also provides a scheme for dynamic graphics display. On a single page, this enables the Content Provider to determine dynamically which Buttons are visible and invisible at any point in time.

[0109] For HDMV BD-ROM content, Movie Objects and the Button Objects contain navigation commands. A Movie Object is executed when the Title associated with the Movie Object begins playback. Movie Object navigation commands are used to manage PlayList playback. While a PlayList is under playback, the state of a Movie Object is maintained. A Button Object is an alternative programming method that is available while the PlayList is under playback and a Button Object is executed by user activation or system timer.

Building an Object Model of a HDMV BD-ROM

[0110] Using processes similar to those described above with respect to interactive multimedia content authored in accordance with the DVD-Video specification, content authored in accordance with the BD-ROM specification utilizing the HDMV framework can be parsed to create an object model. The Index Table can be parsed to identify each of the Movie Objects. The Movie Objects can then be used to insert objects corresponding to the Movie Playlists, Clips, and Button Objects into the object model utilizing the navigation information associated with the Movie Objects and the Button Objects. Once the object model has been constructed, the object model can be utilized to generate an HTML5 page for each Movie Object and an associated JavaScript file that captures the navigation information contained within Movie Objects and the Button Objects. In addition, audio/video/subtitle information in the clips associated with each Movie Object can be extracted, transcoded (optional) and inserted into one or more container files in a manner similar to that outlined above with respect to the extraction of audio/video/subtitle information from VOB files.

Building an Object Model of a BD-J BD-ROM

[0111] The process for building an object model of title with a BD-J object associated with it is similar to the process outlined above with the exception that the content authoring system parses the BD-J object to identify standard objects and translates navigation instructions within the BD-J Java Xlet. In many embodiments, the content authoring system has access to the original source code for the Java Xlet and can construct an object model from the source code. When the original source code is not available, the content authoring system can parse the bytecode of the Java Xlet. In a number of embodiments, standard object libraries are used in the authoring of BD-J Java Xlets and the content authoring system is configured to identify the standard objects in building the object model. Once the object model has been constructed, the object model can be utilized to generate an HTML5 page or a set of HTML5 pages for each BDJ object and an associated JavaScript file that captures the navigation information contained within the BD-J Java Xlet. In addition, audio/